DISTRIBUTORS OF
LOCK SMITH

OMEGA MICROWARE, INC.
222 SO. RIVERSIDE PLAZA
CHICAGO, IL 60606
312-648-1944

THE INSPECTOR by bill sefton
AND OTHER FINE SOFTWARE
AND HARDWARE PRODUCTS

THE COMPLEAT GUIDE TO LOCKSMITH PARAMETERS

This document describes all user-changeable LOCKSMITH parameters in depth. A partial list of these parameters was previously distributed to all Version 4.0/4.1 owners. Also provided here is detailed LOCKSMITH program logic information.

Note---This document is of a highly technical nature, and is intended primarily for the ADVANCED user of LOCKSMITH.

BACKGROUND

When LOCKSMITH was first introduced in January 1981, it would copy almost all disks with no special instructions from the user. Only a few disks required parameter changes.

Alas, those good old days are gone forever. Instead of providing the user with better backup policy, software vendors decided to escalate the battle by developing more complicated (and in some cases, bizarre) protection techniques. Because of the many different techniques now in use, it is likely that many disks will require some input from the user in the form of parameter changes. Omega Microware currently maintains an extensive list of software, along with the LOCKSMITH parameters used to copy each. Some of the entries on this list are user-supplied, and Omega Microware welcomes information from users regarding how to back-up software not already on this list.

OVERVIEW

LOCKSMITH copies disks by reading a track, performing analysis on the data, and writing the track back to the copy disk.

Reading and writing are fairly straightforward functions. The analysis of the track data is by far the most difficult task, and must provide for flexibility.

Many analysis routines (algorithms) are provided within LOCKSMITH. Each algorithm performs a specific function relating to the analysis of track data.

By changing parameters, the user may select, disable, or change the execution order of algorithms. Parameters may also be used to define values to be used by individual algorithms.

ALGORITHMS

The algorithms are numbered from 0 to $23 (all values are in hex), although more algorithms may be added in future versions of LOCKSMITH. During track analysis, algorithms are selected sequentially from a table of algorithm numbers, located from PARM4C-80. As algorithms are selected from this table during analysis, they are displayed on the screen as 2-digit hex numbers in inverse video. Algorithm 00 indicates a null algorithm, which can be used to replace algorithm numbers in the table which the user wants to disable. An FF entry in this table indicates the end of the algorithms to perform.

Currently, the algorithm table contains four separate algorithm sequences, each one terminated by an FF entry. The starting point of the algorithm sequence to be used is defined by PARM25. This parameter contains the index into the algorithm table to be used as the first algorithm of a sequence. For example, if PARM25=00, the algorithm sequence would start at PARM4C. If PARM25=10, the algorithm sequence would start at PARM5C.

The section of algorithm table starting at PARM71 is selected as an algorithm sequence start (instead of PARM4C) when synchronized tracks are chosen.

Algorithms, in addition to performing their specialized function, can return a flag to indicate success or failure. It is possible to indicate that an algorithm is to be performed only if the previous algorithm failed. This may be done by setting the high-order bit of the algorithm number within the algorithm table. For example, an entry of A1 indicates that algorithm 21 is to be performed only if the previous algorithm failed.

DESCRIPTION OF ALGORITHMS

The following is a list of algorithm numbers and the parameters which affect them.

ALG 00 (this algorithm doesn't do much of anything)

ALG 01 (consecutive nibbles to self-sync) changes normal nibbles to self-sync nibbles based on: finding (PARM10) consecutive nibbles in the range (PARM34) to (PARM35), inclusive. For example, if PARM10=0C, PARM34=FE, and PARM35=FF, then algorithm 01 would search for sequences of length 0C nibbles with values FE through FF, and set them to self-sync.

ALG 02 (invalids to self-sync)
sets invalid nibbles (those with 3 or more consecutive zero bits) to self-sync.

ALG 03 (standardize self-sync)
sets all self-sync to (PARM33), which must have high-order bit clear.

ALG 04 (loner self-sync to normal)
set consecutive self-sync strings of less than or equal to (PARM3C) to normal.

ALG 05 (glitch remover)
set consecutive normal nibbles of length less than or equal to (PARM12) to self-sync.

ALG 06 (set self-sync by marker pattern match)
searches for pattern specified by (PARM44-4B), and sets the previous (PARM40) nibbles to self-sync. Values of 00 within the pattern are "don't care" and always match.

ALG 07 (extend bit shifted self-sync)
extends self-sync strings backwards, using the table at (PARM86-A5). This table contains nibble value sequences frequently found to be self-sync.

ALG 08 (reserved for future use)

ALG 09 (trackstart after longest gap)
set trackstart to first normal after longest string of self-sync (gap).

ALG 0A (minimum length self-sync)
extend self-sync strings backwards to minimum length of (PARM2C).

ALG 0B (set self-sync by self-sync pattern match) set self-sync based on multiple-byte pattern match. Pattern is defined at (PARM81-85) and is terminated with a 00 value.

ALG 0C (shorten all gaps)
shorten all gaps (consecutive strings of self-sync) by (PARM41) nibbles if the string length was greater than or equal to (PARM16).

ALG 0D (2 of 3 gap merge)
merges first and second gaps (by setting to self-sync, nibbles between them) if 3 gaps are found within (PARM26) nibbles. (The gaps merged are usually the gap after a data field.)

ALG 0E (trackstart after first self-sync)
sets trackstart to first normal after the first string of self-sync.

ALG 0F (shorten longest gaps)
shorten the longest gap if longer than (PARM2C) by (PARMxx) nibbles. Repeat this procedure (PARMyy) times.
   xx=27 (or 29 if synchronized)
   yy=28 (or 2A if synchronized)

ALG 10 (reserved for future use)

ALG 11 (set failure flag)
same as algorithm 00, but sets the failure flag

ALG 12 (trackstart by marker pattern match)
set trackstart to the first sequence to match pattern at (PARM44-4B) (see ALG 06)

ALG 13 (center of gaps to normal)
leaving 8 self-sync at the start and at the end of a gap, set self-sync in the center of the gap to normal.

ALG 14 (bit-translate to self-sync)
using the bit table at (PARMD9-E8), translate nibbles corresponding to a one-bit to self-sync. Bits in the table represent values for nibbles in the following order: 80,81,82, ... FC,FD,FE,FF

ALG 15 (reserved for future use)

ALG 16 (reserved for future use)

ALG 17 (track-end and compare)
This algorithm searches for a repeat of the track-start beginning at (PARM1D) pages beyond the current track-start. A repeat of the track-start is determined by matching (PARM1E) number of nibbles. If the track size is greater than (PARM1B) pages, an error 2 status code will be issued. Once a track-end is chosen, the first two track images are compared, nibble for nibble. If an unequal nibble compare occurs, a look-ahead of up to (PARM13) nibbles is performed, looking for self-sync. If self-sync is found, the compare failure is ignored. If no self-sync is found during this look-ahead, a counter is incremented for the compare-failure, and this count is checked against (PARM14), which must not be exceeded, or an error 4 status code is issued immediately. The 3rd track image is then used as a tie-breaker to determine which of the 1st or 2nd track images is correct. The exact position in the 3rd track image is found by first finding the approximate location in the 3rd image (by using track length), backing up (PARM11) nibbles, and pattern-matching (PARM32) number of nibbles, while searching through the next (PARM31) number of nibbles. The 1st image is corrected by the tie-breaker nibble. This algorithm returns a success/fail flag.

These algorithms are used to dynamically modify parms. The table at (PARMB6-D8) consists of several sequences of parm modifier entries. Each parm modifier entry consists of a pair of bytes. The 1st byte defines the parm number, and the 2nd byte defines the new parm value. The end of a sequence is indicated by a 00 entry for parm number, and a new sequence begins with the next byte. Algorithm 18 invokes the 1st sequence of parameter modifier entries, algorithm 19 invokes the 2nd sequence, etc. Using these algorithms, parameters may be automatically changed and restored during analysis. The defaults for these algorithms are currently set as follows:

   ALG18 sets 13-sector parms
   ALG19 sets 16-sector parms
   ALG1A sets misc. parms
   ALG1B sets nibble-counting parms

ALG 20 (goto nibble buffer address)
This algorithm is used in conjunction with the nibble editor. This algorithm prompts the user for an address to go to, and the nibble editor cursor is immediately placed at that location (see INVOKING ALGORITHMS FROM THE NIBBLE EDITOR)

ALG 21 (set error code 1)
issues an error 1 status code. It is usually placed in the algorithm table with the high-order bit set, to cause it to execute only when the previous algorithm fails.

ALG 22 (backup trackstart to front of gap)
moves the trackstart pointer backwards to the beginning of the preceeding gap

ALG 23 (set trackstart to longest normal)
sets trackstart to the 1st nibble of the longest sequence of normal nibbles.

PRINTER CONTROL PARMS

(PARM2D) specifies the printer slot, and (PARM2E) is set to 00 if LOCKSMITH is not to generate (CR) at the end of a line, or left at 01 if (CR)'s are to be generated.

MAXIMUM ERROR COUNT PARMS

(PARM01),(PARM02), and (PARM04) are used to specify the number of errors allowed for error codes 1,2, and 4 in automatic error retry mode. If increments of 1/2 tracks are used, (PARM09),(PARM0A), and (PARM0C) are used instead.

NIBBLE-COUNTING PARMS

There are 3 parameters which are used when nibble-count preservation is desired. Setting (PARM36) to 01 turns on nibble-counting. The nibble-count tolerance value, (PARM37), specifies how close to the original disk, the copy must be. When nibble-counting, the track-end pointer is moved up by (PARME9) pages before writing.

PARMS USED FOR SYNCHRONIZING

(PARM22) specifies the track*2 to sync to. This is normally 00, but may be set to any track. (PARM1F) is the length of the nibble sequence to sync with, and (PARMA6-B5) contain the pattern to match when attempting to sync on the sync-track. Values of 00 within the pattern are "don't care" and always match. (PARM23) and (PARM24) are values which can be used to adjust the accuracy of the sync-track routine. They are normally equal, and can be adjusted by increasing the value of one with respect to the other.

PARMS USED TO CONTROL WRITING

(PARM20) contains the lead-in self-sync nibble value. (PARM2F-30) (default is $1A00) number of these lead-in self sync nibbles are written before track data is written, with the exception of synchronized track writing, which is preceeded by (PARM23) lead-in self-sync nibbles. The number of framing bits (1 or 2) is contained in (PARM21). This places the proper number of trailing zero-bits after self sync. (PARM2B) contains the number of the algorithm to be used to shorten the track after an over-write is detected by verify readback failure.

OTHER PARMS

(PARM38) is the number of nibbles to test during verify readback. (PARM39), if set non-zero, shows the hi-res screen during analysis, to provide a graphical representation of analysis. (PARM3A) is used during disk certify. It specifies the maximum size of the track-end glitch. (PARM3B), when set to 01, causes the nibble-editor to be entered for every track, before analysis.

DEBUG PARAMETER

(PARM00) is a special parameter intended for use during LOCKSMITH debugging.
When this parm is set to 11, certain debugging options are enabled. They are:

1. Inspector entry is allowed even with no resident RWTS.
2. Nibble-editor is entered without
   prompting the user for track to read. This allows the previous track to be examined
3. Invoking algorithms from the nibble-editor.
   (see next section)

INVOKING ALGORITHMS FROM NIBBLE-EDITOR

With DEBUG parm set (PARM00=11), the nibble-editor is sensitive to two additional commands. These are control-S and control-A. Control-S invokes LOCKSMITH track-analysis for the track currently in the nibble buffer. Control-A, first allows the user to change parameters by entering the parameter modifier, and after the user has indicated the end of parameter changes with a <CR>, it prompts the user for algorithm number. The user-entered algorithm number is executed immediately, and control is returned to the nibble-editor. In this way, the user can dynamically test the effects of specific LOCKSMITH algorithm sequences when attempting to copy unknown disks. Algorithm 00 can be specified if no processing is to be done. Algorithm 20 is very useful within the nibble editor to rapidly go to a specific address within the nibble buffer.

PARMS AND THEIR DEFAULTS

The following list shows the current default values for parameters of LOCKSMITH version 4.1:

```
00: 00 01 03 01 0A 01 01 01          78: 03 0E A1 17 13 00 00 00
08: 00 01 02 01 03 01 01 01          80: FF D5 AB 00 00 00 FF FE
10: 0C 07 0B 09 0A 78 20 68          88: FD FB F7 EF DF BF FF FC
18: 00 01 08 26 00 16 0D 09          90: F3 CF FF FE F9 E7 9F FE
20: FF 01 00 08 08 00 40 04          98: FE FC F9 F3 E7 CF 9F 00
28: 10 01 10 0F 08 01 01 00          A0: 00 00 00 00 00 00 D5 AA
30: 1A 10 07 7F FE FF 00 00          A8: 96 00 00 AA AA AA AA 00
38: FF 00 0C 00 02 04 02 50          B0: 00 00 00 00 00 00 21 01
40: 06 04 C1 0F D5 AA 00 00          B8: 40 08 2C 08 00 21 02 40
48: 00 00 00 00 00 02 01 06          C0: 06 2C 06 00 3C 08 83 FF
50: 05 0D 07 03 09 A1 17 0C          C8: 00 38 02 1E 02 19 00 12
58: FF 19 02 01 06 05 0D 03          D0: 01 36 01 00 00 00 00 00
60: 09 A1 17 0C FF 1A 02 01          D8: 00 00 00 00 01 00 00 00
68: 0B 05 04 03 09 A1 17 0C          E0: 01 00 01 00 01 01 01 11
70: FF 00 02 01 06 05 0D 07          E8: 5F 00 00 00 00 00 00 00
```